WEST Search History

DATE: Thursday, October 17, 2002

Set Name side by side	Query	Hit Count	Set Name result set
	DB=USPT,PGPB,JPAB,EPAB,DWPI,TDBD; PLUR=YES; OP=ADJ		
L13	L12 NOT l5	0	L13
L12	L11 NOT l8	21	L12
L11	l9 NOT l10	33	L11
L10	L9 and point-to-point	7	L10
L9	L8 or l5	40	L9
L8	((dynamic or dynamically) near4 (partition or partitioning)) and (routing near4 table)	13	L8
L7	(dynamic or dynamically) near4 (partition or partitioning)near8 (routing near4 table)	0	L7
L6	(dynamic or dynamically) near4 (partition or partitioning) near8 (processor or resource) near8 routing	0	L6
L5	(partition or partitioning) near8 (processor or resource) near8 routing	27	L5
L4	L3 and point-to-point	38	L4
L3	L1 and (routing near5 table)	64	L3
L2	L1 and routing	279	L2
L1	(flexible or flexibility or dynamic or dynamically) near8 (partition or partioning or partitionning or partitioned)	5498	L1

END OF SEARCH HISTORY

WEST

Generate Collection

L4: Entry 10 of 38 File: USPT Feb 13, 2001

DOCUMENT-IDENTIFIER: US 6188759 B1
TITLE: Method and apparatus of processing a call in a telecommunications network

Abstract Text (1):
A telecommunications network (10) includes a network processor (16) and a plurality of telecommunications switches (12). Each of the plurality of telecommunications switches (12) receives routing recommendations from the network processor (16) based on congestion data supplied by each of the plurality of telecommunications switches (12) to the network processor (16). Congestion data is collected and supplied for each of the plurality of processors (14) within each of the plurality of telecommunications switches (12) within the telecommunications network (10). Each of the plurality of processors (14) routes telecommunications signals within the telecommunications network (10) according to routing recommendations stored in destination node tables (30) as provided by the network processor (16). The network processor (16) continuously updates the routing recommendations within the destination node tables (30) according to changes in congestion states or configuration states of the telecommunications network (10).

Brief Summary Text (4):
Telecommunications switches are used to route telecommunication traffic through a telecommunication network. The telecommunication switches use routing tables to determine where the telecommunication traffic is to be transferred. Many of these routing tables are static in nature and extremely difficult to update and change. Even those routing tables which might be dynamically updated only provide routing information for the telecommunication switch with which it is associated. However, routing determinations may be performed by any of a number of processors within the telecommunications switch. Therefore, it is desirable to dynamically provide routing information to each of a plurality of processors within a telecommunications switch.

Detailed Description Text (4):
Each telecommunications switch 12 is connected to network processor 16 with fully redundant data links 18a and 18b. Redundant data links 18a and 18b are dedicated point-to-point connections. Redundant data link 18b is provided as a backup in case redundant data link 18a fails. Network processor 16 selects one of redundant data links 18a and 18b as the active link. The active link is the link used by network processor 16 for priming telecommunications switches 12 into dynamically controlled routing operation and exchanging recommendation and congestion data with telecommunication switches 12. The link not selected as the active link is established as a standby link. Network processor 16 establishes the active link first before establishing the standby link.

Detailed Description Text (13):
If the call has been forwarded to DCR switch B, DCR switch C becomes a Handicap Zero destination to DCR switch B. DCR switch B will first try to route the call through a direct trunk group to DCR switch C. If no direct trunk group is available, DCR switch B will look to recommendations made by network processor 16 and within destination node table 34 for tandem routing to DCR switch C. If no tandem routes are available, DCR switch B will refer to switch based fixed route list 36 for alternate routes. If no alternate routes are available, the call is blocked.

Detailed Description Text (15):
If the call proceeds to DCR switch B, DCR switch E becomes a Handicap One destination to DCR switch B. Since there are no direct trunk groups between DCR switch E and DCR switch B, DCR switch B accesses appropriate locations within destination node table 34 for tandem routing recommendations to DCR switch E as determined by network processor 16. If DCR switch D is the recommendation, the call is routed from DCR switch B to DCR switch D over an available direct trunk group. If a direct trunk group is not

available, DCR switch B accesses appropriate locations within switch based fixed route list 36 for alternate routes. If there are no available alternate routes, then the call is blocked.

Detailed Description Text (16):
If the call proceeds to DCR switch D, DCR switch E is a Handicap Zero destination to DCR switch D. DCR switch D will first attempt to route the call over a direct trunk group to DCR switch E. If none is available, DCR switch D will access appropriate locations corresponding to DCR switch E in destination node table 38 for tandem routing recommendations to DCR switch E as determined by network processor 16. If tandem routes are not available, DCR switch D accesses appropriate locations corresponding to DCR switch E in switch based fixed route list 40 for alternate routing. If alternate routing is not available the call is blocked.

Detailed Description Text (18):
Telecommunications switch 12 may also operate in a dual homing mode. The dual homing mode is a switch based routing function which allows calls to be routed on a percentage basis for each country code. The dual homing mode is used for international called party destination numbers. The dual homing mode allows calls to be routed on a percentage basis, in one percent increments, to a virtual DCR switch. A virtual DCR switch consists of two physical DCR destination switches. The virtual DCR switch preferably contains eight direct trunk groups, the number of trunk groups to each physical DCR destination switch within a virtual DCR switch is flexible ranging from zero to eight. Existing routing mechanisms below the trunk group level stay the same. A virtual DCR switch is a non-communicating node with direct trunk groups to each physical DCR destination switch and cannot be recommended as a tandem routing recommendation by network processor 16. If a virtual DCR switch is recommended for tandem routing, such recommendation is ignored. Congestion and data reporting for a virtual DCR switch is the same as for the physical DCR destination switches. For each international call, the call is to be routed on a percentage basis to each of the two physical DCR destinations which are the virtual DCR switch for each country code. Calls are routed based on a route list in an international routing table in the physical DCR destination switches.

Detailed Description Text (53):
As described, the dynamically controlled routing function is partitioned among and within many processors of each telecommunications switch 12. There is not one single processor doing everything to perform the DCR task, but tens to hundreds of processors working together and coordinating with each other to accomplish the DCR function. Each processor performs a unique and specific function that cumulatively provides a comprehensive DCR capability. Coordination is performed through interprocessor messaging such that processors can keep track of what other processors are doing.

L4: Entry 26 of 38 File: USPT Sep 1, 1998

DOCUMENT-IDENTIFIER: US 5802043 A
TITLE: Transport architecture and network elements

Abstract Text (1):
A novel transport architecture and network elements use nodes in a ring or virtual ring configuration. Information is exchanged among the nodes in a continuous series of containers carried in a superframe. A container is assigned to a destination node and is divided into partitions which are allocated for source nodes. Therefore, when a container destined to a particular destination node arrives at a node and if this node has information to send to the destination node, it fills the allocated partition within the container with information destined to the destination node. The sizes of the superframe, containers and partitions are all variable dynamically or statically.

Detailed Description Text (6):
These parameters can be fixed at provisioning time, or they can be adjusted dynamically to respond to changing traffic demands. For example, if the bandwidth of the domain with 5 nodes is fixed at 50 Mb/s, and a 10 frame multiplex frame is allocated with two containers per domain node, then each destination node can receive a maximum of 10 Mb/s. If each of the other nodes were only allowed to fill one quarter of each container, then the bandwidth between any two nodes is 2.5 Mb/s. An access protocol operating between all the nodes would allow partitions within a container to be adjusted dynamically. For example, one node could be given 5 Mb/s if two other nodes only require 1.25 Mb/s. Similarly, if a destination node requires more than 10 Mb/s, it could be allocated 3 containers per multiplex frame (15 Mb/s) if another destination node only requires one container (5 Mb/s). The granularity of this adjustment (5 Mb/s in this example) can be changed by going to a longer multiplex frame. This adjustment can also be made dynamically. Finally, all of the numbers in this example could be tripled by going to a 150 Mb/s domain. A cost/performance trade-off determines which parameters should be adjusted dynamically and how fast this adjustment needs to be.

Detailed Description Text (11):
Referring to FIG. 6, container packers operate in a master/slave mode, with every packer being the master for the containers destined to the local domain node, and slave for each of the other containers. The master sources a container with partitions allocated to each of the other packers to fill in. The master can dynamically adjust the partitions in subsequent containers based on traffic demands. In the figure, therefore, the master packer (node A) sends out containers with the partitions defined for the slaves (nodes B, C and D) to fill in their information. In a further embodiment, rather than sending out empty containers, which would be wasteful of capacity, this bandwidth is utilized for additional capacity from the master to each of the slaves.

Detailed Description Text (16):
c) reads addresses of incoming frames from the adaptation function and determines from a routing table (established in shared memory) the next network hop and thus the appropriate container;

Detailed Description Text (31):
It is also possible that the stevedores can be organized hierarchically, either to limit routing table size in lower layer stevedores, or to take advantage of hierarchical addressing, as seen in FIG. 9. The stevedores on a single path ring can be part of a single addressing domain. A set of path rings, and consequently all the stevedores attached to them, can have a peer relationship. A higher level path backbone ring can be used to interconnect the path rings at the lower level. If the addressing used by the networking functions is hierarchical, then the networking functions on the backbone ring only need look at the portion of the address identifying the lower layer domain to get across the backbone path ring. Similarly, networking functions on the

lower layer rings simply forward traffic to the networking function on the backbone ring if they do not recognize the portion of the address that specifies the lower layer domain. Otherwise, they use the portion of the address that specifies the individual networking function to get across the lower layer ring.

Detailed Description Text (35):
As seen in the above description, this embodiment provides a means for SONET-based transport networks to provide connections at virtually any bandwidth, and to allow that bandwidth to vary in real-time according to instantaneous traffic demands. It also allows many such connections to share transport capacity on a statistical basis. This increases the efficiency of bandwidth utilization on the transport network. Telecommunications transport networks based on traditional SONET only allow point-to-point path layer connections at one of three granularities, 1.7 Mb/s, 50 Mb/s, or n.times.150 Mb/s. These bandwidths are reserved across the entire network regardless of instantaneous traffic demands.

L4: Entry 38 of 38 File: USPT Apr 9, 1996

DOCUMENT-IDENTIFIER: US 5506840 A
TITLE: Asynchronous switching node and routing logic means for a switching element used therein

Brief Summary Text (43):
Whilst the above new routing type may for point-to-point connections be theoretically applied for all cells that are handled by a switching element, this would result in a required capacity of the routing memory that is prohibitively large for it to be integrated on the same integrated circuit as the other circuits of the switching element with current state of the art technology. Indeed, the routing memory would approximately have to have a memory location for each group of output ports supported by the switching network.

Brief Summary Text (44):
Since a separate integration of the routing memory would lead to serious commercial and technical drawbacks, a second circuit for point-to-point connections, already known from the last referenced application and using combinatorial logic, is included in the routing logic means. The capacity of the routing memory can then be reduced to only those applications strictly requiring it. In this way the capacity of the routing memory can be so reduced as to make it feasible for this memory to be integrated on a same integrated circuit as the rest of the switching element.

Detailed Description Text (11):
FIG. 2 illustrates the definition of a distribution tree and a branch. It is a diagram showing the switching elements of a network SN1 effecting only point-to-point routing. The situation of point-to-multipoint routing or multicasting is discussed later. FIG. 2 shows a distribution tree for cells arriving at an input A and addressed to a group LG comprising four output ports B, C, D, E.

Detailed Description Text (77):
The routing logic means RL derives from an internal routing label, henceforth also called self-routing tag SRT, applied on a like named input terminal a group routing signal GL also applied on a like named terminal. GL comprises 14 bits to each of which a distinct routing group is associated as described in detail in the above European patent application EP/A1/0446493. Each of the mentioned routing groups include a number of the 16 outlets of the switching element, the composition of each routing group being specified in a routing group table (not shown). The routing logic means can be seen externally as selecting based on the internal routing label, i.e. the self routing tag SRT, one or more of the mentioned routing groups by setting those bits of GL which correspond to these selected routing groups. Based hereon, i.e. on the output signal GL, the switching element can forward the information cell(s) associated to the mentioned SRT to one outlet of each selected routing group, such an outlet being chosen substantially randomly from amongst all the outlets within its routing group. It is to be noted that the above international application WO91/02420 describes in particular that the above is implemented by putting the information cell(s) in a queue corresponding to a selected routing group, each routing group being associated to one such queue.

Detailed Description Text (78):
The routing group table in the present embodiment specifies for instance 8 routing groups each including 2 outlets, 4 routing groups of 4 outlets and 2 routing groups of 8 outlets. These routing groups are so chosen that each outlet is included in exactly one routing group of a specific size.

Detailed Description Text (86):
It can be easily seen that, due to the latter routing function, the flexibility of the routing is no longer confined to the predefined routing groups. Indeed, a set of

routing groups allows the information cell(s), associated to an SRT for which this set is activated, to be distributed over a number of outlets which are not as such included in a routing group and which do not comprise all outlets of the switching element. This set is moreover activated for all information cells associated to a specific SRT and such distribution can thus obviously be tailored to suit specific connections only. Sets can also be easily added/deleted in response to the emergence/disappearance of their associated connections without effecting the other connections. In this way it will also be avoided that an unfeasible amount of routing groups needs to be defined in the routing group table since the above mentioned sets will not be needed all simultaneously and since the basic routing groups, which can only be redefined affecting all connections passing through the switching element, naturally must only reflect the interconnection of a switching element within its switching network.

Detailed Description Text (91):
It is to be noted that an advantageous feature of the present invention is that the described point-to-point and point-to-multipoint routing functions can be simply achieved via one way, i.e. the routing memory RM and the selection means SEL. A further advantage is that, thanks to the routing group size signal, some form of controlled distribution, i.e. amongst a controlled number of outlets, can be achieved.

Detailed Description Text (92):
A further known routing function which can be performed by the present routing logic means is point-to-point routing based on an output address contained as such in the routing control address RCA. This function is performed by the combinatorial routing logic CRL for which reference is made to the mentioned European patent application 0446493A1. The mentioned output address will be in its simplest form the digital representation of the reference number of the routing group in the present routing logic means but can, as described in the latter application, also be a variable number of bits of RCA which have to be interpreted in accordance with information present in the routing indicator RI as will be briefly described hereafter.

Detailed Description Text (96):
It is to be noted that it was just the above described way of point-to-point routing selecting a routing group via combinatorial logic based on an output address contained in the routing control address RCA which resulted in this type of routing being confined to the predefined routing groups. The reason for the inclusion of CRL also in the present routing logic means lies in the low hardware cost of this way of routing. Indeed, when all information cells were to be routed via RM, the latter would become prohibitively large since substantially every connection to be supported by the switching element would require a separate memory location in RM. Such a large memory cannot be realized on a single integrated circuit. It is therefore very advantageous to include CRL in the routing logic means since such inclusion reduces the number of memory locations of RM thus making it possible for the switching element to be integrated on a single electronic chip.

Detailed Description Text (97):
As will be explained with reference to the switching network of FIG. 11 a hybrid form of routing where in some stages use is made of the output port address whilst in other stages the routing memory is used may also lead to an advantageous decrease of memory capacity needed for the above new way of routing. It has moreover to be noted in this respect that the routing groups, in case the output address routing via CRL is used, generally will not be of the partitioned nature described above in order not to limit the flexibility of this routing. The present invention however already provides enough flexibility so that, in principle, another definition than the partitioned one is no longer needed.

Detailed Description Text (119):
With respect to the above latter embodiment of a switching node and a routing logic means according to the present invention, it has to be noted that these embodiments practically constrain the new routing type to point-to-point connections only. However, such a restriction is clearly only an attribute of these latter embodiments and by no means of the invention as such as can be clearly appreciated from the switching node and routing logic means embodiment discussed with reference to FIG. 8.

WEST

Generate Collection

L12: Entry 1 of 21 File: PGPB Nov 1, 2001

DOCUMENT-IDENTIFIER: US 20010037435 A1
TITLE: Distributed address mapping and routing table mechanism that supports flexible configuration and partitioning in a modular switch-based, shared-memory multiprocessor computer system

Summary of Invention Paragraph (16):
[0015] The nodes of the multiprocessor system may be configured as a single address space or a plurality of independent address spaces, each associated with a hard partition of the system. In a system having a single address space, each routing table is programmed such that a specific address or processor ID range may be assigned to at most one node or processor, which may have one physical connection to the switch fabric. However, in a system having a plurality of hard partitions, disparate routing tables are utilized and a specific address or processor ID range may be assigned to one node or processor or to multiple nodes or processors, which may have common or disparate physical connections to the switch fabric. Here, if ranges are assigned to one node or processor, the nodes associated with the routing tables are included in a common hard partition. On the other hand, if ranges are assigned to multiple nodes or processors, the nodes associated with the routing tables are included in disparate partitions. In any case, all routing tables are programmed such that all hard partitions are mutually exclusive.

CLAIMS:

12. The method of claim 11 wherein, for each hard partition, the routing table entries corresponding to a selected node that is part of the respective hard partition are programmed such that a processor of the selected node is assigned processor ID zero and the respective routing table entries are valid.

21. The method of claim 20 wherein, for each hard partition, the routing table entries corresponding to a selected node that is part of the respective hard partition are programmed such that a processor of the selected node is assigned processor ID zero and the respective routing table entries are valid.

27. The modular, shared memory multiprocessor system of claim 26 wherein the logical IDs and valid bits of the routing table entries are further configured such that, for each hard partition, a given processor has an ID of "0".

WEST

Generate Collection

L12: Entry 12 of 21 File: USPT Oct 19, 1999

DOCUMENT-IDENTIFIER: US 5970232 A
TITLE: Router table lookup mechanism

Abstract Text (1):
A multiprocessor computer system includes processing element nodes interconnected by physical communication links in a n-dimensional topology, which includes at least two global partitions. Routers route messages between processing element nodes and include ports for receiving and sending messages, and lookup tables having a local router table having directions for routing between processor element nodes within a global partition, and a global router table having directions for routing between processor element nodes located in different global partitions. The directions from the local table are selected for routing from the next router along a given route if the current processing element node is in a destination global partition or if the current processing element node is one plus or minus hop from reaching the destination global partition and the route is exiting on a port that routes to the destination global partition, else the directions from the global router table are selected for routing from the next router.

Brief Summary Text (19):
The present invention provides a method and a multiprocessor computer system including a plurality of processing element nodes. Each processing element node has at least one processor and memory. Physical communication links interconnect the processing element nodes in a n-dimensional topology, which includes at least two global partitions of processing element nodes. Routers route messages between the plurality of processing element nodes on the physical communication links. Each router including ports for receiving and sending messages, and lookup tables associated to ports and holding entries having directions for routing from a next router along a given route. Each lookup table includes a local router table having directions for routing between processor clement nodes within a global partition, and a global router table having directions for routing between processor element nodes located in different global partitions. The directions from the local table are selected for routing from the next router if the current processing element node is in a destination global partition or if the current processing element node is one plus or minus hop from reaching the destination global partition and the route is exiting on a port that routes to the destination global partition, else the directions from the global router table are selected for routing from the next router.

CLAIMS:

1. A multiprocessor computer system comprising:

a plurality of processing element nodes, each processing element node having at least one processor and memory;

physical communication links interconnecting the processing element nodes in a n-dimensional topology, which includes at least two global partitions of processing element nodes;

routers for routing messages between the plurality of processing element nodes on the physical communication links, each router including:

ports for receiving and sending messages, and

lookup tables associated to ports and holding entries having directions for routing from a next router along a given route, each lookup table including a local router table having directions for routing between processor element nodes within a global partition, and a global router table having directions for routing between processor

element nodes located in different global partitions, wherein the directions from the local table are selected for routing from the next router if a current processing element node is in a destination global partition or if the current processing element node is one plus or minus hop from reaching the destination global partition and the given route is exiting on a port that routes to the destination global partition, else the directions from the global router table are selected for routing from the next router.

L12: Entry 14 of 21 File: USPT Jun 17, 1997

DOCUMENT-IDENTIFIER: US 5640327 A
TITLE: Apparatus and method for partitioning resources for interconnections

Brief Summary Text (2):
The present invention pertains to the field of interconnections for coupling together system components. More particularly, the present invention pertains to an apparatus and method for the partitioning, placing, and routing of electrical components with fixed and variable resources.

Detailed Description Text (13):
There are different constraints and requirements during the partition process associated with the different interconnection problems. For example, in the Field Programmable Gate Array (FPGA) case, where the amount of logic and routing resources are very fixed, the partition consideration needs to account for both the boundary crossing requirements to match with the available capacity at each partition level and the logic, routing, and access resources available within each partitioned bin. For the Masked Gate Array (MGA) case, there can be some trade-offs among different levels in sizing up the boundary crossing needs at each partition level, even though the total routing resources is fixed. For the Multiple Chips Modules (MCM) case, the capacity at a certain level of partition may be fixed, while at some other levels they may be flexible. For example, in the MCM case, at the substrate level, the partition objective minimizes the amount of connections at the substrate while guaranteeing the success at the chips level.

Detailed Description Text (28):
Referring back to FIG. 7, each of the four partitioned bins 440, 450, 460, and 470 contains one-fourth the total logic area from the parent bin, which is proportional to the array area under study. Each boundary crossing span linearly over two bins, which is proportional to the length of the array under study. Similarly, in FIG. 8, the size of each of four partitioned bins for the next level is found to be inversely proportional to the array area of the parent bin, while the boundary crossing distance is inversely proportional to the length of the array of the parent bin. If one views each boundary crossing pattern as having potentially a maximum connection span of the combined lengths of the two abutting bins and overlays those patterns for all levels of partitions, one can arrive at an estimated bound for routing resources required over the total system. Thus, in the currently preferred embodiment, the partitioning process of the present invention divides the interconnect problem into multiple levels of two-dimensional routing resources models by utilizing a divide-by-four partition. At each level of the partition process, there is a set of routing patterns with routing length span approximately twice of those in the next level of partition and approximately half of those in the previous level.

Detailed Description Text (29):
Now, a modified example of FIG. 6 is used to illustrate this innovation as applied to a Masked Gate Array application. FIG. 11A shows a sample routing structure for a sea-of-gates masked gate array. As indicated by the horizontal and vertical grid lines two levels of routing are available. Furthermore, FIG. 11 shows a top view, wherein the system is first divided into four parts (i.e., four bins). The same design example of FIG. 6 is now used to illustrate the partitioning method as applied to a Masked Gate Array application. Instead of a four bits datapath (as shown in the FPGA case), the design is now considered to be 8-Bits. At the top system level, there are several input and output signals that must be accessed through the peripheral of the system boundary. Specifically, these I/O signals include thirty-two data input signals D[31:0], eight register output signals Q[7:0], and five common control signals S0, S1, CLK, RN, L. There are eighty horizontal grids and eighty-eight vertical grids at the top level (i.e., four copies of FIG. 11A). This is sufficient to handle forty-five signals. FIG. 11B shows the partitioning of the 8-bits data path into four bins. For the first level

of partition, each of the four bins 1110-1140 has sufficient routing resources to guarantee internal routing success within each bin. The constraint or the limiting factor is the amount of logic that can be grouped together within each bin. Specifically, each bin has enough capacity for two bits of logic, but not for three bits. Thus, during the first step of partition, each bin contains logic components for two bits datapath. Specifically, Bin 1110 contains bits 0-1 of FIG. 6, Bin 1120, bits 2-3; Bin 1130, bits 4-5; and Bin 1140, bits 6-7. Once the components partition are completed, the next step is to look at those signals that crossed the bin boundaries. In this example, the signals are: signal CLK crossing boundaries 1111, 1112, and 1114; signal RN crossing boundaries 1112, 1113, and 1114; signal SO crossing boundaries 1112, 1113, and 1114; signal S1 crossing boundaries 1111, 1112, and 1114; signal L crossing boundaries 1111, 1112, and 1114; signal Q[1]-SI[2] crossing boundary 1111; signal Q[3]-SI[4] crossing boundary 1112; and finally, signal Q[5]-SI[6] crossing boundary 1113.

Detailed Description Text (35):
There is a major difference between FPGA and MGA during the top down partition and bottom up implementation process. The routing resources for FPGA are mostly fixed at each level of the partition process. In contrast, for MGA applications, the total resources are fixed but can be flexible at each level of partition by trading off resources from one level to the other. For fixed capacity cases (e.g., FPGA, Masked Gate Arrays, etc.), wherein the number of component placement sites and amount of routing resources are fixed, failure in any step to place and route the components within any bin at any level means that the implementation of the system can not be completed using the above method. There are two alternatives when such a failure occurs. One approach is to transfer the design to a larger resources part and repeat the process until it is successful. The other approach is to adapt an iterative procedure whereby the Top-Down Process is revisited with a more refined capacity estimation. An iterative analysis on alternative boundary-crossing estimates is then performed.

CLAIMS:

8. The method of claim 1, wherein said step of optimizing an assignment of each component to one of said plurality of first partitions is a function of partition capacity, routing resources, and access resources.

19. The apparatus as set forth in claim 11, wherein said means for optimizing an assignment of each component to a plurality of first partitions and means for establishing subsequent levels of partitions optimize as a function of partition capacity, routing resources, and access resources.

20. A method for partitioning a surface area of a programmable logic device into four partitions corresponding to a level of routing resources of said programmable logic device, said method comprising the steps of:

dividing said surface area in two in a first axis;

dividing said surface area in two in a second axis, wherein said second axis intersects said first axis perpendicularly and said surface area is divided into said four partitions;

determining said routing resources that are required to implement a circuit design based on a routing pattern of the routing resources, said step of determining said routing resources determining extra routing resources needed to route connections between a first component located within a first partition of the four partitions to a second component located within a second partition of the four partitions, wherein the first and second partitions are diagonally opposed partitions and the routing resources needed to provide L-shaped connections are included;

optimizing said routing resources in relation to a component capacity of said programmable logic device.

21. The method of claim 20, further comprising the steps of determining a number of components of said programmable logic device, a number of routing resources that span across boundaries of said four partitions, and input/output routing accesses; wherein said steps of dividing said surface area in two in a first axis, dividing said surface area in two in a second axis, determining and optimizing are not performed if the number of components, the number of routing resources that span boundaries and

input/output routing accesses exceeds a device capacity.

23. The method of claim 21, wherein a span of a routing resource that crosses one of said boundaries is proportional to lengths of two of said four partitions.